ISD Firewall Evaluation

Introduction.

The last few years have seen an explosion in the importance of computer networks, both inside and outside of State Government. Along with the increasing importance of these computer assets comes an increasing need to protect them, both to prevent unauthorized use of these assets and to assure their continuing, reliable availability. In today's world this protection effort is greatly enhanced by the deployment of network Firewalls to help us control access to our network resources. Just as a guard may aid us in implementing the Security Policy that regulates access to our physical plant, a Firewall can aid us in implementing our Network Security Policy.

ISD has evaluated nine Firewalls for the Windows NT 4.0 operating system. The level of in-house experience with administering Windows NT, as opposed to UNIX, dictated NT's choice as the Firewall platform. Part 1 of this document provides a non-technical description of what a Firewall is and is not, the reasons one may need a Firewall, the different types of Firewalls, and the way in which your Security Policy will dictate which type of Firewall is right for you. Part I is brief, six pages, and concludes by summarizing the results of the evaluation process and reveals ISD's suggested vendors for each type of Firewall. Part II describes, in a non-technical manner, the evaluation process and evaluation criteria and how this process resulted in the recommended vendor list. A detailed technical report of the evaluation is presented in a companion document.

Part 1 - ISD Firewall Evaluation

What is a Firewall? Why do I need one?

Firewalls are needed for much the same reason we lock doors: to control access to a set of valued resources. In the case of a Firewall the resource to be protected is a local area network (LAN) and we wish to control access from some other network, or networks, be it the global Internet or some other portion of the State's network. The central concept here is that of controlling access - a firewall is not a magic bullet that confers immunity from any and all sorts of evil, it is simply one of the ways we implement the policy that regulates access to our LAN. For example, imagine that we have decided that our Network Security Policy has the following goals:

1) Anyone on our internal network can access anything on either the Internet or the rest of the State network; *i.e.* anyone on the internal network can access anything on the external network.
2) Anyone on the external network can send e-mail to the internal network, but no other contact initiated from the outside is permitted.
3) We wish to log, or record, all other attempts made to access our internal network.

This is a fairly common security policy[1], one that allows unrestricted access from our network to the outside world, but tightly regulates access from the outside world to our internal net. A Firewall would help us maintain this policy by being a chokepoint through which all network communication between our internal net and the outside world must pass. In this role as a sentry, our Firewall examines all the network traffic that passes through and only allows the passage of traffic that fits our security policy.

However, it must be realized that a Firewall is not a universal cure-all for security concerns, but only one of the ways in which we implement our Security Policy. A Firewall cannot protect our network from the ravages of a virus that was carried into our internal net on a floppy disk, nor can it do its job if any other path into or out of our internal network exists, such as a modem pool, simply because it will never see the traffic flowing via this other route.

From what can Firewalls protect me?

Permitting access from external networks to your trusted, internal network can cause you harm in one of two ways: 1) someone, or something, can cause your computer to behave in ways that you do not want or 2) the data that travels to and from your trusted network is visible to prying eyes, much like a post card. We shall deal with each of these in turn.

Crashes and Hacks

The security concern that gets the most press is the break-in, also known as the hack. As one would think based on the name, a break-in has occurred when someone has gained

[1] This is simply an example Security Policy; others might permit unregulated access to the State network and limit, or even prohibit, access to the Internet, while yet others would permit mail, Web, and FTP originating from the Internet into the internal network.

unauthorized access to your machine. In the Windows world, such unauthorized access would include reading the contents of your hard drive, or your e-mail, or sending some e-mail while impersonating you, or perhaps damaging, or modifying, your data. None of these are good things, and a Firewall can help control this risk by limiting the number of avenues that can be used to gain access to your machine. Note that we said help control: a Firewall can and should prevent people from the outside network from gaining access to your machine, but it provides no protection from attacks that originate on the internal network. Historically, unauthorized access to a computer is more likely to be an inside job.

A second, and distressingly more common, type of attack is the denial of service. Here the goal of the attack is not to obtain access to your computer, but to deny you access to your own computer. This type of attack consists of sending your computer some precisely crafted data that causes your machine to crash merely as a result of receiving the malicious data.[2] These attacks are much more likely to originate outside your network and so a Firewall can provide superb protection simply by preventing the outsider from being able to reach your computer.

Encryption

Under normal circumstances the data sent or received by your computer is not encrypted, and can thus be read by any or all of the computers it passes by in route to its destination. For many applications this is of little concern, but when sharing data with a remote site or sending confidential e-mail, the lack of encryption can be a major security risk. Firewalls are uniquely placed to minimize this risk simply by encrypting data as it leaves your trusted, internal network. For example, if you wish to share data with a site hundreds of miles away, and they too have a firewall, your firewall can encrypt the data as it leaves and their firewall can decrypt the data as it arrives[3].

How do Firewalls work?

Firewalls work by examining all of the network traffic that passes through them, allowing traffic that adheres to our Security Policy to pass and blocking everything else. For a Firewall to do it's job, our internal network must be configured with a single point of entry through which all network traffic must pass. When described abstractly as a regulator of network traffic, a Firewall's job appears quite simple. Unfortunately, like most things in the real world, our simplification has hidden some features that influence, or even control, how Firewalls do their job. To appreciate this, and to fully understand the tradeoffs involved in choosing a particular Firewall, we must more precisely define what we mean by network traffic.

---

[2] At this writing, the most recent, widespread attack of this sort took place the first week of March, 1998 and was directed at Microsoft Windows machines. It was an Internet based attack in which Educational and Federal government installations were targeted and was extraordinarily successful, causing Windows machines from California to Maine to crash. The people responsible have not been caught.

[3] It is also possible to implement encryption without the remote site having a firewall. In this case, the remote, or destination, machine needs to have appropriate software installed so that it can do the encryption/decryption process for itself.

What is network traffic? What is a packet?

When we sit down and view a web page, or read our e-mail, or transfer a file, we tend to think of each web page, or each mail message, or each file transfer as a single ***session***. This is indeed true. When we view a web page our computer contacts a particular ***service*** (in our example, a World Wide Web service) on a particular remote machine. If that service is not offered by that remote machine, we are notified, and the session ends; if that service is offered, a ***connection*** is established, the web page is transferred to our computer, and then both the session and the connection are terminated.

However, at a lower level, the network level, the concepts of session and connection are not applicable. Each single session, whether it be a request for a single web page, or a single e-mail message, or single file transfer - each single session - consists of tens if not thousands of chunks, or pieces, of information termed ***network packets*** that travel in each direction, both to and from your computer. Each packet contains a portion of the data you are sending or receiving as well as address information that serves to route the packet from its source to its destination; it is important to note that the ***service*** is a portion of the address. We can think of a single packet as a post card, with a return address, a destination address, and the data, or message, to be sent. Each single session consisting of a web page that we view, or an e-mail message that we read, or a file that we transfer consists of a flurry of these packets travelling both to and from your computer and the remote machine.

How do Firewalls process network traffic?

Just as we can view network traffic at the session level or the packet level, we can group Firewalls into two types based on the way they view network traffic. The first type, Packet Filters and Stateful, Multi-Layer Inspection Firewalls, view network traffic as consisting of packets and have only a limited notion of sessions. This type of Firewall regulates network traffic by examining the address information in either a single packet or a short packet stream, and permitting or denying the passage of the packet or packet stream. The second type of Firewall, Application and Circuit Level Gateways, view network traffic as consisting of sessions, and uses both the address information and the data present in each packet to allow or disallow the formation of a session. Each type has its strengths and weaknesses.

Packet Filters and Stateful,Multi-Layer Firewalls

The easiest way to understand how a Packet Filter works is by way of our postcard analogy. Imagine that we control a building that has a guard posted at its single door. We are very security conscious, so we have decided that our Security Policy will only permit the use of postcards for communication between the inside and the outside of our building. Each post card must contain both the address of the sender and the recipient as well as the message; note that either the sender or the recipient may be inside our building. At this point we merely wish to limit who may correspond with whom so we have provided our guard an access list – some people on the inside may receive post cards, and others not. To implement this security policy, our guard examines the address information on each post card, consults his access list, and either allows the post card to enter (or leave) our building or not. A Packet Filter operates on network packets in exactly the same manner that our guard operates on post cards: it

examines the source and destination address information present in each packet and either allows the packet to pass or not.

If all we care about is who is sending and receiving postcards, then this Security Policy and its implementation are perfectly satisfactory. However, this may prove to be insufficient for our needs so we may place further limitations on the flow of postcards. For example, we may require that a correspondence, or a flow of post cards in both directions, can only be initiated from inside our building. This presents our guard with a problem: individual post cards contain no information that our guard can use to determine who initiated the correspondence. Our guard, however, is a clever sort and he solves this problem by modifying the nature of the access list. In addition to an access list that we provide, our guard begins to record who has initiated a correspondence, and with whom. Now our guard can examine a single post card, consult the access list and his list of ongoing correspondences, and allow only those post cards that are part of an ongoing correspondence to enter the building. Stateful,Multi-Layer Inspection (SMLI) Firewalls do exactly this.

An SMLI Firewall incorporates the session concept by maintaining a list of open connections. This permits an SMLI Firewall to have one set of rules for establishing connections, and a second set of rules applicable for established connections. By distinguishing between established and non-established connections an SMLI Firewall can implement our tightened Security Policy.

Four of the Firewalls evaluated for ISD fall into the SMLI category: Checkpoint System's Firewall-1, Guardian for NT, Cisco System's PIX, and the Elron firewall.

Application and Circuit Level Gateways

Let us return to our post card analogy and our guard, who was examining the addressing of each post card and using an access list and a list of open correspondences to determine which post cards could enter or leave our building. Even though we have control over who can receive post cards, we have become concerned about the contents of some of the messages: someone inside the building may be using the post cards to transmit information that we wish to remain confidential. So we now require that, in addition to examining the addressing of each post card, our guard reads each message, and further, actually makes a copy of the post card and lets only the copy, and not the original, enter or leave the building. The decision to permit a post card into our building is now based both on the address information and the contents of the message. It is this approach that is taken by Application Gateway Firewalls.

Packet Filters and SMLI Firewalls are largely concerned with regulating the connections formed between individual computers and services, and pay little attention to the data flowing via that connection. Thus, an SMLI Firewall will be able to determine if my computer has permission to form a session with a particular Web server, but will not be able to regulate what I do with that connection. Because an Application Level Gateway does monitor the data that flows via a connection, it can exert such control. To appreciate this distinction, imagine that we wish to implement a Security Policy in which machines on our internal, trusted network are allowed to view any and all web pages, but as a precaution against computer viruses, denies

them the ability to download any executable file to their computers. An Application Gateway can easily implement such a Security Policy while an SMLI Firewall would at best have difficulties, and at worst be totally unable to implement our desired policy.

Unfortunately, as is often the case in the real world, all is not sunshine and roses for Application Gateways. The increased control provided by Application Gateways comes at a definite, two-fold cost. The first price lies in speed: the examination of data passing back and forth slows things down even if the examination is trivial. Simply providing this capability, even if it is not used, extracts a performance penalty. The second cost lies in decreased flexibility. To monitor the data flowing via every connection through it, an application level gateway must understand each and every protocol that flows through it. In terms of our post card analogy, a protocol is the equivalent of the language used to write the post card, and our guard must understand each and every language in which a post card may be written. An Application Gateway accomplishes this through the use of dedicated, special purpose programs termed proxies for each protocol that passes through the Firewall. When new protocols emerge (such as RealPlayer or SQLNet) an application level gateway cannot handle them until a proxy has been written. For this last reason, no commercial firewalls are simple Application Gateways.

Circuit Level Gateways attempt to correct this second short-coming through the addition of a general purpose proxy. These are reminiscent of the features of SMLI Firewalls in that they merely examine the source and destination of a connection in determining whether to allow or deny the session, and do not look at the data flowing through that connection. However, to coexist with the Application Level Gateways they must suffer the same performance cost. Their purpose is not to increase the performance of the Firewall, but to increase its flexibility.

Five firewalls evaluated for ISD fall into the Application/Circuit level Gateway class: Altavista Firewall, Cisco's Centri Firewall, TIS's Gauntlet, Raptor Systems EagleNT and IBM Firewall for NT.

Which type of Firewall is best?

There is no one answer to this question, rather, the best type of Firewall depends upon the Security Policy you wish to implement. If your primary security concern lies in controlling which machines can reach your trusted, internal network from the outside, an SMLI Firewall is the best choice. It provides great flexibility in specifying which machines can communicate with each other, will install into your existing network in a truly transparent manner, and is very fast. However, an SMLI Firewall will provide little, if any, control over how the connections between your trusted network and the outside world are used. To provide such control one must examine the data that is carried within each network packet, something that, by and large, an SMLI Firewall does not do by its very design.[4]

On the other hand, if your Security Policy required more control over outgoing connections, including the ability to deny access to particular Web pages, or perhaps to limit,

[4] There are extensions to SMLI Firewalls that add on the ability to examine the data portion of each packet, but they often require a second machine to perform the examination.

or prohibit, the downloading of files, an SMLI Firewall would be a poor choice. In this case, your needs are better satisfied by an Application/Circuit Level Gateway type of Firewall. There is a tradeoff implicit in this choice – Application Gateways are less flexible and tend to be slower[5], but they do provide much greater control over network communication.

Evaluation Results - Summary.

In Part II of this document we turn to a more detailed, though non-technical, evaluation of the Firewalls in each category. Evaluation criteria include ease of administration, cost, security and remote management capabilities. All criteria were evaluated by installation of the Firewall into a small test network that permitted both the installation and administration process to be evaluated, as well as permitting various forms of performance and stress testing.

Nine Windows NT 4.0 Firewall products were evaluated. Four of the nine were SMLI Firewalls: Checkpoint System's Firewall-1, Guardian, Cisco System's PIX, and the Elron Firewall. Both Cisco's PIX and the Elron Firewall have design limitations that prevent remote administration; because ISD requires the ability to remotely administer any Firewall deployed, these products are unacceptable. Of the remaining two, Guardian and Firewall-1, Checkpoint's Firewall-1 is ISD's recommended solution. While both products provide excellent SMLI based Firewall protection, Firewall-1's superior encryption and remote administration capabilities gave it the edge.

The remaining five Firewalls were in the Application Gateway category: the Altavista Firewall, TIS's Gauntlet for NT, Raptor System's EagleNT, the IBM Firewall for NT, and Cisco System's Centri Firewall. Three of these products, the Altavista Firewall, the IBM Firewall and Cisco's Centri displayed rather large stability problems. In fact, none of the three were capable of completing the evaluation tests. Of the remaining two, Gauntlet and EagleNT, Raptor Systems EagleNT is ISD's recommended solution. As with the SMLI Firewalls, the key features in which EagleNT was superior were remote management and encryption.

Both Checkpoint System's Firewall-1 and Raptor System's EagleNT are superb products and both provide an excellent security solution. As described above, which is best suited for your needs will depend on your Security Policy.

[5] With the speeds of computers in the late 1990's, the speed decrease imposed by Application Gateways is only a factor for networks that function at speeds above those of 10Base-T.

Part II – ISD Firewall Evaluation

To this point we have been considering Firewalls as rather abstract objects, and have broken them into two types – The SMLI Firewalls and the Application Level Gateways – and have indicated that it is the Security Policy to be implemented that determines which type is best for a particular LAN. It remains to determine how well each of the Firewalls do their jobs. The following SMLI Firewalls were evaluated: Checkpoint System's Firewall-1, Guardian for NT, Cisco System's PIX, and the Elron firewall. The following Application Gateway Firewalls were evaluated: Altavista Firewall, Cisco's Centri Firewall, TIS's Gauntlet, Raptor System's EagleNT and the IBM Firewall for NT. Windows NT was chosen as the Firewall platform because ISD will be responsible for the administration of any Firewall installed and ISD has a much greater level of in-house administration expertise with Windows NT. Because of the inherent differences between the two types of Firewalls, the evaluation results for each type are presented separately, though each type was subjected to the same evaluation tests.

SMLI Firewalls

Of the four SMLI Firewalls evaluated, only two were able to complete the full range of tests. The two unacceptable Firewalls were the Elron Firewall and Cisco's PIX, and both suffered from administration problems. For both Firewalls, remote management via the external network is impossible; it is expressly excluded by the product design. While from a security standpoint, this design decision is eminently justifiable, and perhaps even desirable, it is not acceptable here: the support of Firewalls that may be scattered throughout the State-wide network requires that administration from a single, central point be possible via the Firewall's external network.

This leaves the Guardian Firewall and Checkpoint System's Firewall-1. These products are remarkably similar in appearance, leading one to suspect that Guardian (a relative newcomer) is imitating Firewall-1, the industry leader. Because of their similarity, we shall evaluate them side by side, using one to point out the pluses and minuses of the other.

Adherence to defined Security Policy is one clear requirement of any Firewall. By this we mean that a) only those connections expressly permitted by the Security Policy should be capable of being formed and b) only packets that are part of an allowed or established connection should get past the Firewall; *i.e.* there should be no packet leakage. Neither Guardian nor Firewall-1 displayed any problems with respect to non-allowed connections. Each product allowed all of the connections defined in the Security Policy, and no others. However, packet leakage proved more problematical. We care about packet leakage because the first step in attacking computer networks is obtaining an accurate map of the network you wish to attack. Ideally (from the attacker's viewpoint) such a map would contain the IP addresses, OS types, and network services being run by every machine on the target network. Packets that leak through a Firewall can be used to construct just such a map.

Packet leakage was tested by attempting to send a variety of packet types through the Firewall under a variety of conditions (*i.e.* with no connections open, a small number of connections, a large number of connections) and using packet capturing programs to observe

which packets made it through the Firewall. Guardian proved superior to Firewall-1 in preventing packet leakage. Guardian permitted the passage of only those packets destined for services that were explicitly enabled in the Security Policy, while Firewall-1 permitted some packet types to reach all ports, regardless of Security Policy[6], and in some cases permitted a return packet to reach the "attacker". This is a definite security hole in Firewall-1: by using these unusual packet types, an attacker can map the machines behind your Firewall, determining how many there are, and which services each machine supports. No Firewall should allow this.

On the other hand, Firewall-1 was superior to Guardian in reporting such mapping attempts. While Guardian successfully blocked the use of odd packets to map a network, it did not report the attempt even when instructed to do so. Firewall-1, though it allowed some of the mapping attempts to succeed, reported each and every attempt.

Protection from Denial of Service (DoS) attacks. A distressing trend of the late 1990's lies in the increase of Denial of Service (DoS) attacks, an attack whose goal is to deny you access to your own computer resources. There are two types of DoS attacks. In the first type, the attacker simply attempts to flood your network with bogus connection requests, causing either the network or individual computers or both to bog down and eventually become totally unresponsive. Both Guardian and Firewall-1 provide protection against such attacks. It should be noted that this protection applies only to the network that the Firewall protects. It may remain possible to flood the network outside of your Firewall, and thus deny you access to the resources on the outside net.

In the second type of attack, the attacker is attempting to send your computer a precisely crafted sequence of network packets that causes your machine to crash or hang. This attack requires two things: 1) a bug of some sort in your machines Operating system that the attacker can exploit and 2) the ability to send network packets that reach your machine. Over the last nine months there have been a variety of such attacks targeting Microsoft Windows machines, and we would like our Firewall to protect the machines behind it from such attacks. Because Microsoft has made available software patches, or hotfixes, that prevent the exploitation of all currently known DoS attacks we wish to test our Firewalls with two configurations: one in which the appropriate patches have been applied to the Firewall's NT operating system, and one in which the patches have not yet been applied.

In both of these configurations, with and without patches, both Guardian and Firewall-1 blocked the attack if the Firewall was configured to block attempts to access the service at which the attack was directed. That is, if the Firewall was blocking incoming Telnet requests, and we mounted a telnet based attack, both Guardian and Firewall-1 protected themselves and the machines behind the Firewall on the protected network. The situation gets murkier if the attack is directed at a service that the Firewalls are configured to allow from the external network.

---

[6] Internet-like services, such as the World Wide Web, Mail, Telnet, etc. utilize what are known as ports. If, for example, one is trying to reach the web server on the machine whose IP address is 1.2.3.4, the complete address specification is 1.2.3.4.80, where the 1.2.3.4 specifies which machine is being contacted, and the 80 specifies the port on that machine. Port 80 is the standard WWW port.

If the Firewall machine was vulnerable to the attack (*i.e.* we had not installed the patches available from Microsoft) and an attack was directed at an allowed service on the Firewall machine itself, the Firewall machine crashed. While this does protect the machines behind the Firewall from the DoS attack, this is not a desirable solution. If the Firewall machine itself was patched, and the machines behind the Firewall were not, both Guardian and Firewall-1 were capable of protecting the machines behind them from some DoS attacks, but not all. This is to be expected with an SMLI Firewall and presents a definite administrative problem. Fully protecting the machines behind the Firewall will require applying the needed patches to each and every machine behind the Firewall. Moreover, one would expect that as new types of DoS attacks are mounted, that either the Firewall itself or the machines behind them will prove to be vulnerable.

Ease of Administration. Both Guardian and Firewall-1 have straightforward, easy to use Graphical User Interfaces (GUIs) for administration. Because the interfaces were so similar, what follows applies to both products.

The installation and setup of each firewall was straightforward and quick. Each consists of two components, a Firewall Agent or Module that performs that actual regulation of network traffic, and a Firewall Manager that is used to define network objects (*i.e.* the IP address of the Firewall itself, the network addresses of the networks to be protected, etc.) and the set of rules that forms the Security Policy; the Security Policy is then loaded onto the Firewall Module. These Security Policy rules consist of a source and destination IP address or IP address range, the type of service, and the action to perform – accept or deny the packet. Thus the administrator can control which machines can communicate and via which services this communication is allowed. When the administrator is defining the rules that makeup the Security Policy care must be taken with their ordering. When a packet arrives at the Firewall, it is examined and compared to the rules that have been defined as the Security Policy beginning with the rule numbered 1, and continuing sequentially until the ***first*** match is found. At that point the packet is either passed on or dropped as specified by that rule. Thus, the order of the rules matters. For Administrators that are unfamiliar with packet based Firewalls, this can cause misconfigurations.

All in all, both Firewalls were very easy to manage and install.

Remote Administration. Because ISD will be responsible for the administration of any Firewall, remote administration is a requirement for any product. Both Guardian and Firewall-1 have excellent remote administration capabilities; in fact, one does not have to install the Firewall manager on the box that hosts the Firewall Module. All functionality described above is available via remote management. One advantage that Guardian possesses lies in the automatic encryption of any remote management session. Unfortunately, Guardian does not limit the machines that are capable of remote management (*i.e.* there is no IP address based access control list) and access is controlled by a single password. Multiple Administrators may be connected simultaneously, but only the first to connect can modify either the network object definitions or the Security Policy's rule base. Other Administrators are limited to read-only access.

Guardian has one major problem with remote administration – the network objects and rules used to create the Security Policy are stored only on the machine whose Firewall manager created them, and are **not** available to other remote management stations, nor can other Remote Firewall Managers view the Network Security Policy currently installed on the Firewall module. This would clearly create a nightmarish situation if multiple workstations are used to create Security Policies.

Firewall-1 does not automatically encrypt remote management sessions, but does have an IP address based access control list that limits the machines that can form a management connection. Encryption can be easily added by installing the SecuRemote client on either a WindowsNT or Windows95 machine. (See the section on Encryption, below.) In addition, Firewall-1 requires both a username and a password to create a management session, and permits the definition of multiple Administrators that can have either full control, or read-only access. If multiple, full control Administrators attempt to log in, all but the first are limited to read-only access. Finally, unlike Guardian, each Remote Firewall Manager obtains both the network objects and rules that are currently installed on the Firewall module.

In short, if there is to be a single Firewall management station there is little to choose between Guardian and Firewall-1. However, as the State's needs may well require multiple, fully capable Firewall Management stations, Firewall-1 is clearly superior.

Transparency. For the purposes of this evaluation, transparency has two components: first, the installation of the Firewall should require no change to the existing networks, and the Firewall's design should not be the factor that determines which services can be provided to the internal network. Both Guardian and Firewall-1 are truly transparent by both criteria – this is one of the great advantages of the packet based type of Firewall. Out of the box, both products provide built-in SMLI rules that support a wide variety of network protocols, though Firewall-1 supports a larger number. For each Firewall, the Administrator can define new rules that would permit the Firewall to support new protocols. However, Guardian's ability to define new rules is limited to simple packet filtering – the Administrator cannot write Stateful-Multi Layer Inspection rules to support the new protocols in a secure manner. Firewall-1, on the other hand, includes its proprietary INSPECT language that can be used to create new SMLI rules. This is a big advantage for Firewall-1 in that it will allow a vendor that has developed a new protocol to write a Firewall-1 rule that supports that protocol in a secure manner.

Both Guardian and Firewall-1 provide superb transparency. Firewall-1's inclusion of the INSPECT language gives it an advantage in places where unusual, or novel, protocols will be deployed.

Logging and Report Generation. Guardian and Firewall-1 each use the rules that compose the Security Policy to control the level of logging that is performed. Two levels of logging are supported by both products: 1) a simple log entry that specifies the time the packet was observed, which rule number it matched, the source and destination IP addresses and ports, and 2) an accounting record for the session in which that packet was a part. The simple log entry is very useful in debugging Firewall problems, but would quickly lead to extraordinarily

large (Gigabyte size) log files if enabled during normal operations. The accounting records are a different matter. Both Firewall-1 and Guardian create accounting records that include the source and destination IP addresses, who initiated the session, session duration and bytes transferred, and both permit the viewing of these records while the session is ongoing. However, Guardian updates these records more frequently and permits the administrator to suspend individual connections. This last feature may or may not be of use.

Neither product is very strong at manipulation of either the log files or accounting files. Both will automatically move log files to a permanent location (on a scheduled basis) and each provide the capability to view both the log and accounting files. Within the viewer, the administrator can filter which records are viewed, export the file as a delimited text file suitable for importing into another application, or print the log, or a portion of it. This is the full extent of both products logging/report generation capabilities.

Encryption and Virtual Private Networks (VPNs). Under normal circumstances the data in any network communication is not encrypted, meaning that as the data travels from its source to its destination, it is readable by any and all machines passed by in route. Virtual Private Networking (VPN) technology eliminates this worry by automatically creating an encrypted communication channel between properly configured machines. These machines can either be two firewalls, so that all data that travels between the networks that each Firewall protects would be encrypted, or between a Firewall and a stand-alone remote computer. Checkpoint's Firewall-1 possesses a clear-cut advantage over Guardian when encryption enters the picture. Guardian is capable of forming a VPN with another Guardian Firewall, but not with anything else, including stand-alone remote clients.

Firewall-1 can form VPNs with another Firewall-1 module and with any machine that has SecuRemote installed; the ability of Firewall-1 to form VPNs with other Firewalls was not tested due to equipment limitations. However, SecuRemote was installed and tested on a Windows NT machine on the outside network and what follows applies to SecuRemote.

SecuRemote's installation was straightforward, and the product proved stable and transparent under the test conditions. To use SecuRemote with Firewall-1 the Firewall Administrator has to perform three tasks: 1) define an encryption domain – this is a range of IP addresses behind the Firewall for which Firewall-1 will provide encryption services, 2) define a set of users, with usernames and passwords, and 3) set up an appropriate rule in the Security Policy to grant encrypted access. For example, imagine that the user named joe needs encrypted access to the entire internal network from the outside. The Firewall Administrator would first define the entire protected network as the Firewall's encryption domain, then create a user account for joe. This account can be just a Firewall account (*i.e.* all account information is stored locally on the Firewall) or some authentication server can be used, such as Radius. Finally, after the account and encryption zone have been created, the Administrator creates a special rule that defines from where users of SecuRemote may connect to the protected network. Firewall setup is then complete.

On the SecuRemote client, one simply installs the software, and then enters a set of IP addresses or IP address ranges for which SecuRemote should be invoked. That is it. Then, whenever that client machine attempts to contact an IP address on SecuRemote's list,

SecuRemote and Firewall-1 form an encrypted channel and the user is prompted for his SecuRemote username and password. After that, all things proceed normally. It is this mechanism by which Firewall-1 provides encryption for its Remote Management function.

Speed. This is another of the advantages of packet based Firewalls – they are fast. While our testing environment precluded a real world test, with tens or hundreds of clients accessing the Firewall, two types of benchmarks were performed. One was a simple FTP transfer of a large file and the other was a network performance benchmark suite, netperf. Both benchmarks provide a measure of raw network speed and so, theoretically, we can obtain some measure of Firewall overhead by comparing performance with and without the Firewall in place.

Unfortunately, we were not able to place enough of a load on these types of Firewalls to notice any decrement in speed; *i.e.* in our test set up the only factor limiting performance was the bandwidth of the physical network, and not anything that we asked the Firewall to do. This result does **not** mean that the Guardian and Firewall-1 Firewalls will not slow down your network. It merely means that for simple byte transfers, Guardian and Firewall-1 can keep up with a 10 megabit ethernet network.

Our test procedures did not include heavy use of encryption, nor a heavy load of accounting procedures simply because we did not have at our disposal the number of computers that such tests would require. Based on reports available on the Internet, Firewall-1 will slow down under conditions of a large number of encrypted connections (on the order of a 50 to 100) and under conditions of a heavy accounting responsibility.

With these caveats aside, it should be noted that both products have been designed to handle hundreds of protected clients at speeds higher than 10 megabit ethernet can provide. If network bandwidth increases to the 100 megabit range, these firewalls will become a bottleneck.

Stability is one clear cut requirement for any Firewall - it must be up and functioning correctly 24 hours a day, seven days a week, 365 days a year under any and all conditions. Unfortunately, no short term testing procedure can provide much information about long term stability. To provide some indication of Firewall stability, both Guardian and Firewall-1 were subjected to two types of stress tests: 1) the power to the Firewall machine was turned off, and then back on to verify that the Firewall would function normally after a spontaneous reboot and 2) large byte transfers were performed continually for a periods up to sixteen hours, and Firewall performance was monitored. Both Guardian and Firewall-1 performed superbly under both tests. Following a spontaneous reboot, both products came up and ran correctly without any administrative intervention, and both products were capable of maintaining high speed byte transfers (~ eight Megabits/second) for periods up to sixteen hours.

The only potential stability problems observed with either product relate to the Denial of Service attacks described above. If the Firewall's Windows NT 4.0 Operating System had not been adequately patched, both products were vulnerable to DoS attacks directed against Windows NT. Both products functioned normally under all other circumstances.

Cost. Waiting for vendor information.

SMLI Firewalls – Putting it all together. Both Guardian and Firewall-1 are excellent products and choosing between them is a tough call. Ultimately, Firewall-1's advantages for an enterprise-wide network give it the edge. The advantages Firewall-1 offer a large scale network are most evident in its Remote Administration capabilities and its encryption support for mobile clients. Firewall-1's design enables multiple administration sites for multiple Firewalls, while Guardian clearly envisions a single management console for all installed Firewall Modules. Moreover, Guardian offers encryption only between two Guardian Firewalls, thus precluding the use of encryption for stand-alone remote clients. Firewall-1's SecuRemote offers seamless encryption for such clients.

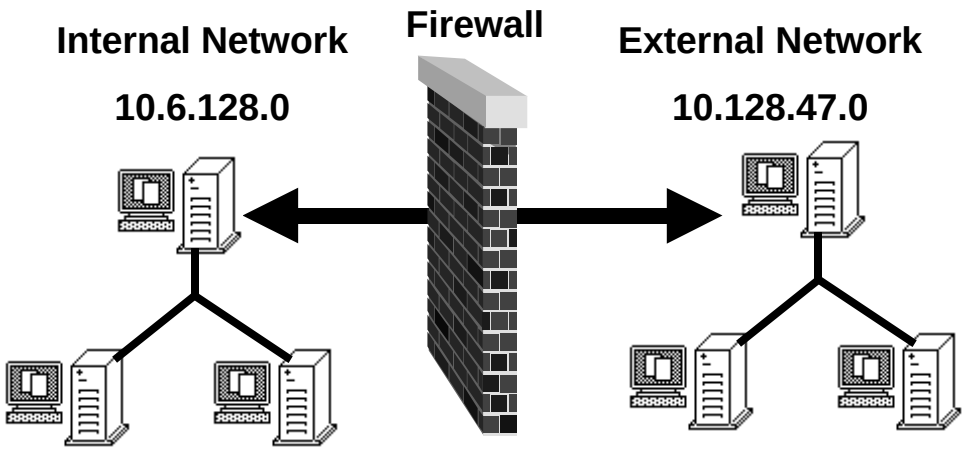
Firewall-1 does have one significant disadvantage: it does permit some mapping of the networks that it protects through the use of odd network packets. This would be a major concern if the product was to provide the sole source of isolation between the State network and the Internet. However, if Firewall-1 is deployed in a State Agency setting, all such mapping attempts from the Internet are already being blocked by the State's Internet Firewall. The extra-protection provided by the Internet Firewall lessens the concern over this flaw in Firewall-1.

Application Gateway Firewalls

Five Application Gateway Firewalls were included in this evaluation: Altavista Firewall, Cisco's Centri Firewall, TIS's Gauntlet, Raptor Systems EagleNT, IBM Firewall for NT. Three of these products, the IBM Firewall, Cisco's Centri Firewall, and the Altavista Firewall, displayed such glaring difficulties that they were not fully evaluated. On two separate occasions the IBM Firewall for NT simply ceased functioning and permitted all packets to pass. Whatever triggered these events was not logged and no notification appeared on the Management console. Cisco's Centri Firewall will not run with Windows NT 4.0, Service Pack 3 and the associated hotfixes installed, so it remained vulnerable to the numerous Denial of Service attacks that have appeared in the last nine months. And finally, while the Altavista Firewall did not crash when subjected to the suite of Denial of Service attacks, its behavior was most unpredictable: on some occasions it would simply cease to function, blocking all packets, on other occasions the entire machine would lock up some minutes after the cessation of the attack. In addition, on two occasions the machine spontaneously rebooted. These stability problems render the IBM Firewall for NT, the Altavista Firewall, and Cisco's Centri Firewall unacceptable.

This leaves TIS's Gauntlet and Raptor's EagleNT; both of these products contain a rich set of Application Level Gateways as well as a packet filtering capability. It should be noted that for both products that enabling packet filtering produces a significant decrease in security; this is a point to which we shall return. We will compare Gauntlet and EagleNT side by side, as well as draw comparisons to the SMLI Firewalls.

Figure 1.



Adherence to defined Security Policy is one clear requirement of any Firewall. By this we mean that a) only those connections expressly permitted by the Security Policy should be capable of being formed and b) only packets that are part of an allowed or established connection should get past the Firewall; *i.e.* there should be no packet leakage. We care about packet leakage because the first step in attacking computer networks is obtaining an accurate map of the network you wish to attack. Ideally (from the attacker's viewpoint) such a map would contain the IP addresses, OS types, and network services being run by every machine on the target network. Packets that leak through a Firewall can be used to construct just such a map.

Neither Gauntlet nor EagleNT displayed any problems with respect to non-allowed connections. Each product allowed all of the connections defined in the Security Policy, and no others.

With the SMLI Firewalls packet leakage is a major security concern. With Gauntlet and EagleNT this is a concern only if the packet filtering feature is enabled. In the default configuration for each product packet filtering is disabled and packet leakage is, by design, impossible. To appreciate this difference between SMLI Firewalls and Application Level Gateways consider the network diagrammed in Figure 1 and what happens when someone on the outside network (the 10.128.47.0 network) attempts to open a web page that is on a server on the inside network with IP address 10.6.128.22. If the Firewall is an SMLI type it will examine the packets that form this request, and if the Security Policy permits the outside network to reach the web server at 10.6.128.22 the packets will pass and the web page will be transferred to the outside computer. Now imagine that the firewall is a standard Application Gateway. Regardless of our Security Policy, this request will be rejected because an Application Gateway **always** rejects attempts to communicate directly with the machines on its protected, or internal, network. If we wanted to set up an internal Web server on the machine with IP address 10.6.128.22 we would first have to configure our Application Gateway Firewall so that it knows there is an allowed web server at 10.6.128.22. Then, the people on the network outside the application Gateway would request a web page from the "web server" at

10.128.47.1 – that is, they would request the web page from the Firewall itself. The Firewall would then pass this request on to the true web server and return the data to the outside machine. Thus, by virtue of its design, an Application Gateway blocks **all** requests to communicate directly with the machines it protects and the packet leakage problem with SMLI firewalls cannot exist. For both Gauntlet and EagleNT this was verified using the same techniques as used for the SMLI Firewalls.

This absolute protection is absent if one enables the packet filtering mechanisms on either Gauntlet or EagleNT. If, in the example described above, we used packet filters to permit the outside network to reach the web server at 10.6.128.22, these packets would flow directly through the firewall, bypassing the application gateway. In this case, all of the well known problems of packet filtering apply, and packet leakage is almost a certainty.

Both Gauntlet and EagleNT are immune to packet leakage type problems **if and only if** they are configured without packet filtering. If packet filtering is enabled, they display greater packet leakage problems than an SMLI Firewall.

Protection from Denial of Service (DoS) attacks. A distressing trend of the late 1990's lies in the increase of Denial of Service (DoS) attacks, an attack whose goal is to deny you access to your own computer resources. There are two types of DoS attacks. In the first type, the attacker simply attempts to flood your network with bogus connection requests, causing either the network or individual computers or both to bog down and eventually become totally unresponsive. In the second, a stream of malicious data is sent to the target computer in hopes of exploiting a bug in the target's OS and causing it to crash.

The same design feature that provides Application Gateways with immunity to packet leakage, that of total isolation of the internal protected network from the outside world, also provides immunity to any and all DoS attacks. If the internal machine cannot be reached, it cannot be attacked. Thus if all the needed hotfixes are applied to the machine hosting the Application Gateway, both the Firewall itself and the machines behind it are immune to Denial of Service Attacks. This immunity was verified using the same procedures As with SMLI Firewalls for both Gauntlet and EagleNT.

***Administration*.** Both EagleNT and Gauntlet have straightforward interfaces, though each could be a bit more intuitive. Installation of the software is adequately documented and involves installation of the Firewall Modules themselves and a separate GUI Firewall Manager. For each product the installation and initial configuration are straightforward; the only potential stumbling block applies to both products and lies in Domain Name Service (DNS) configuration. DNS is the protocol that permits people to use the names of computers (*e.g.* www.state.nd.us) as opposed to their IP address; DNS is the means by which your computer discovers the IP address associated with the name www.state.nd.us. Clearly, any Firewall must make provisions for its clients to use DNS. The solution used by SMLI firewalls, which merely allow DNS requests to pass, is not available to Application Gateways like EagleNT and Gauntlet: no direct contact between inside and outside networks is permitted. Both Gauntlet and EagleNT solve this problem by setting up the Firewall box itself as a forwarding name server, though they use different mechanisms. Both require that the Firewall box itself have the

loopback address specified as the DNS server in the Windows NT Network Properties, but EagleNT uses a built-in DNS for all DNS requests. Gauntlet requires the installation of Microsoft's DNS server as a forwarding name server.

Both products have GUIs for Firewall Management and both are not as intuitive as they might be. One source of possible confusion lies in the use of the term "rule" in both the administration interface and product documentation. Because these products consist of two distinct portions, the Application Gateway portion and the Packet Filter portion, the term rule does not always have the same meaning. That being said, both interfaces were eminently usable once the Administrator adapts to the two different portions of each product.

Remote Administration. This is a major problem for the Gauntlet Firewall. Remote administration of Gauntlet requires that the directory tree on the Firewall be a Microsoft Networking share accessible to the remote management station. The massive security hole that this produces renders remote management functionally impossible. TIS agrees with that assessment and the next version of the product is supposed to address this shortcoming.

Conversely, Remote management is a strength of EagleNT. As with the SMLI Firewalls evaluated, the Firewall Management software need not be installed on the machine hosting the Firewall itself. Full management capability is available remotely, with access controlled both by an IP based access control list and a password. All management sessions are automatically encrypted and only one manager may be logged in at any one time.

Transparency. A loss of transparency is one of the inevitable tradeoffs one makes when using an Application Gateway based Firewall. These tradeoffs can be broken into two types that we will call Transparent Access and Transparent Integration; we shall deal with each in turn.

Transparent Access. Because an application Gateway uses a special purpose program, or proxy, to pass traffic through the firewall, there is an inevitable decrease in the numbers and types of applications that can be run across the Firewall. Both Gauntlet and EagleNT provide a rich set of proxy's that support services including telnet, ftp, HTTP (both Secure Socket Layer (SSL) and non-SSL), Simple Mail Transfer Protocol (SMTP), Post Office Protocol (POP) mail, Microsoft Networking, RealAudio and Video, Gopher, NNTP (news), NTP (network time protocol), and DNS. SQL*Net is reported to work with both Gauntlet and EagleNT, though this was not tested. Finally, any protocol that opens only well-defined ports (*i.e.* the port numbers are known in advance) can be easily added using what is known as a Generic Proxy.

If a protocol cannot be passed via one of the Application Gateways, each product requires the administrator to define a packet filter that will permit the protocol to pass. Each product warns the user **in very strong terms** not to do this. The reasons are simple – opening up a packet filter to support a protocol that cannot be supported via the Generic Proxy will open a large security hole. If your needs require such a solution, you would be better served by using an SMLI Firewall.

Transparent Integration. Many Application Gateway type Firewalls require extensive modifications to the clients and/or network they are protecting. Examples of these

modifications include installing new software on the client machines or redesigning your network to place a DNS server behind the Firewall. If a DNS server does exist behind the Firewall, neither EagleNT nor Gauntlet require a single change in the network they are protecting. However, in the absence of an internal name server, both require a single modification to their client machines: each client machine must be configured so that the Firewall itself is the machine the clients use to resolve DNS requests. Thus, the use of either of these products will require a visit to each of the protected machines so that DNS services can be reconfigured. This is a cost that must be calculated when considering either of these products.

Logging and Report Generation. Just as Transparency is a traditional problem area for Application Gateways, Logging and Report Generation are areas of traditional strength. Gauntlet shows off this strength in its ability to automatically generate reports detailing the usage of each of its proxies. Thus, one can automatically schedule reports either for printing or e-mail distribution detailing telnet, ftp, e-mail, web use, etc. It should be noted that of the products included in this evaluation Gauntlet was the only one to include this ability.

EagleNT's logging capabilities, though much better than either of the SMLI Firewalls, were rather disappointing. A single log file is maintained, and the administrator uses the log viewer to manually select the types of log records to view. The logging is very complete, but As with the SMLI Firewalls, some sort of add on is required to generate reports.

Encryption. While report generation is a clear strength of Gauntlet, encryption is totally lacking in the product. As with Remote Administration, TIS is planning on adding this to the next release of Gauntlet.

EagleNT incorporated encryption in a manner very similar to Firewall-1. Firewall-to-Firewall encryption is possible, though was not tested here. Remote clients gain access to encrypted communication through the installation of software on the remote client, and the definition of a set of IP addresses for which encryption is to be enabled. When such a connection is required, the Firewall requires the remote user to authenticate himself with a username and password. Successful authentication yields the encrypted session.

Speed. As with the SMLI Firewalls, our testing setup did not permit a robust measurement of the speed of these Firewalls. Both products were more than adequate to keep up with 10 megabit ethernet speeds.

Stability is one clear cut requirement for any Firewall - it must be up and functioning correctly 24 hours a day, 7 days a week, 365 days a year under any and all conditions. Unfortunately, no short term testing procedure can provide much information about long term stability. To provide some indication of Firewall stability, both Gauntlet and EagleNT were subjected to two types of stress tests: 1) the power to the Firewall machine was turned off, and then back on to verify that the Firewall would function normally after a spontaneous reboot and 2) large byte transfers were performed continually for a periods up to sixteen hours, and Firewall performance was monitored. Both Gauntlet and EagleNT performed superbly under both tests. Following a spontaneous reboot, both products came up and ran correctly without any

administrative intervention, and both products were capable of maintaining high speed byte transfers (~ eight Megabits/second) for periods up to sixteen hours.

The only potential stability problems observed with either product relate to the Denial of Service attacks described above. If the Firewall's Windows NT 4.0 Operating System had not been adequately patched, both products were vulnerable to DoS attacks directed against Windows NT. Both products functioned normally under all other circumstances.

Cost. Waiting for information from Vendor.

Application Gateways – Putting it all together. EagleNT's support of Remote Administration and Encryption give it clear cut advantages over Gauntlet. While both products performed their security functions superbly, and Gauntlet was the only product included in this evaluation that offered robust Automatic Report Generation, it's inability to perform remote administration in a secure manner make it ultimately unsuitable for the State's needs. Moreover, Gauntlet's lack of encryption support would prevent its use in the formation of encrypted channels from the machine's of remote users. Together, these two features lead us to endorse Raptor's EagleNT as the Application Gateway Firewall solution.